

Mzf2wav.

Jeroen F. J. Laros

September 11, 2003

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | Introduction. | 3 |
| 2 | Doing something with the MZ. | 3 |
| 2.1 | Mzf2raw. | 3 |
| 2.2 | Mzf2wav. | 3 |
| 2.3 | Later versions of Mzf2wav. | 4 |
| 2.4 | Operation modes. | 4 |
| 2.4.1 | Non-redundant writing mode. | 4 |
| 2.4.2 | Redundant writing mode. | 4 |
| 2.4.3 | Turbo writing mode. | 4 |
| 2.5 | Speeds. | 4 |
| 2.5.1 | Corrections. | 5 |
| 2.5.2 | Polarity. | 5 |
| 2.5.3 | Defaults. | 5 |
| 3 | Release. | 5 |
| 4 | See also. | 5 |
| 5 | License. | 5 |
| 6 | References and credits. | 6 |

1 Introduction.

I have just purchased both an MZ-700 and an MZ-800. I've been looking for one of them for years. At last I found an MZ-800 on an internet site, and a few days after buying this one, someone contacted me (via the same internet site) and sold me an MZ-700. What luck!

2 Doing something with the MZ.

So.. I had two obsolete computers. I tested both and carefully stalled the MZ-700. The MZ-800 I connected to my TV and I started to search on the internet for transfer programs. The first thing I wanted was to reliably transfer data and found that the easiest way was to convert an MZF file to WAV and then record this on a tape, which can be read by the MZ. I found a compiled Java program which does just this, but it didn't work under Linux. At this point I probably should have upgraded to the latest Java version, but I thought it would be nice if I wrote my own program. Also the absence of source code made the program virtually useless (at least for my purposes).

2.1 Mzf2raw.

Under Windows 95 I converted a few MZF files to WAV with the Java program I found and closely looked at the output. After a few hours of staring at these files, I saw some patterns emerging. Next I wrote a small C program which did exactly the same thing as the Java program. Then I found the fantastic website: www.sharpmz.org, which would have spared me a few hours of staring (the only thing I didn't find there is that each byte is accompanied by a long pulse). At this site I found how the MZ stores it's information on tape. This is significantly different from the way I was using, so I extended my program to make a RAW file which is according to all conventions. I recommend using this option if you want to store a program on a tape permanently, it will store a copy of the header and a copy of the file to make a tape more reliable. It is also possible to vary the speed (but only for the fast conversion mode).

So, the work was done. All I had to do was convert to RAW and then with a program named sox I placed a WAV header in front of it and I had my WAV.

2.2 Mzf2wav.

When I mailed my program to the owner of sharpmz.org, he told me that most people use Windows NT. Compiling my code for NT was a piece of cake, however, there was one problem. I don't know of a simple command line tool for NT to convert a RAW file to a WAV. The solution was to do it myself. Mzf2wav was born.

2.3 Later versions of Mzf2wav.

The later versions of Mzf2wav are modified versions of Mzput, another program of mine. This is because Mzput has far more functionality and the code is more structured.

2.4 Operation modes.

2.4.1 Non-redundant writing mode.

This mode, also referred to as fast writing mode is the simplest way to write a file. In this mode the long gap (a gap is a sort of marker) is only 4000 pulses and the short gap is 5000 pulses. Both header and MZF body are written once.

2.4.2 Redundant writing mode.

This mode is also referred to as conventional writing mode. This is the safest and slowest way to write an image. As far as I know everything is done according to the Sharp MZ series conventions: A long gap is 22000 pulses, a short gap is 11000 pulses. The header and body are written twice, so in case of read errors, a copy can be used. If you want to store your program on a real cassette, this is the preferred method.

2.4.3 Turbo writing mode.

This mode was not my idea. I found it in the TransManager package, but this is how I implemented it: First we write a turbo loader using non-redundant writing mode, then we alter the writing speed, and finally we write the MZF also using non-redundant writing mode. I don't take credit for the turbo loader, I just copied it (and that's about the only thing I didn't write myself). I did disassemble the turbo loader by hand and tried to comment on it, but I couldn't find good documentation, so there are a lot of question marks in the comment.

2.5 Speeds.

Five speeds are implemented at this time:

- Speed 0: this is the normal speed with normal waveforms.
- Speed 1: this is 'normal' speed with fast waveforms (see the documentation of Mzput for more details).
- Speed 2: this is 2x speed with normal waveforms.
- Speed 3: this is 3x speed with normal waveforms.
- Speed 4: this is 3x speed with fast waveforms.

These speeds can be set for all operation modes, although most combinations will not work, except in specific cases. The most commonly used speeds are: speed 0 or 1 for normal writing modes and speed 2 or higher for turbo writing

mode. If, however you need to write multiple images you might want to use speed 3 or 4 with non-redundant writing mode. Unless you have a very good sound card, speeds other than 0 and 2 will probably fail.

2.5.1 Corrections.

Because the waveforms used by speeds 1 and 4 are quite minimal, I suspect they will not always work. So I implemented two corrections for these speeds. You can fiddle with them if you like. These speeds are actually only for direct communication. I doubt whether your sound card is able to make these waveforms, but as mentioned earlier, since Mzput has this functionality, this program also has it.

2.5.2 Polarity.

I don't know if this is useful, but the -p option reverses the polarity. I never needed it, but since the MZ has one, I figured my program also needed one.

2.5.3 Defaults.

Mzf2wav will default to non-redundant writing mode with speed 0. You can vary the initial speed mode with the -i command line option, the turbo speed mode with the -t command line option and the method with either the -c (conventional), -s (fast) or -w (turbo) options. Furthermore you can correct speeds 1 and 4 with the -1 and -2 command line options.

3 Release.

This release includes the source code, a Linux compiled binary and a Windows compiled binary, a README and this document.

4 See also.

If you're interested you might want to take a look at Mzput. Mzf2wav and Mzput are closely related.

5 License.

This program is freeware and may be used without paying any registration fees. It may be distributed freely provided it is done so using the original, unmodified version. Usage of parts of the source code is granted, provided the author is referenced. For private use only. Re-selling or any commercial use of this program or parts of it is strictly forbidden. The author is not responsible for any damage or data loss as a result of using this program.

6 References and credits.

The turbo loader was copied from Transmanager, which itself is based on work by Miroslav Nemecek. You can download the software from:

<http://mzunity.wz.cz/old/Hardware.htm>

Additional information about the MZ hardware was provided by Karl-Heinz Mau. He also stimulated me to write a w32 version. See:

<http://www.sharpmz.org/index.html>

for almost everything about the Sharp series.